

设计实验室 4

撞墙

6.01 – 2011 年秋季

目标:

在[设计实验室 2](#)，您开发了一个“保持距离”的大脑，即使墙壁移动，也能将机器人定位在墙壁前方的固定距离。在这个实验室中，我们开发了该系统的信号和系统模型，并使用该模型来了解大脑/机器人系统的性能。有了这个模型，我们将能够回答关键的性能问题，例如：响应是快还是慢？系统是否超调或振荡？

这将分三步进行研究：

- 开发墙壁探测器系统的差分方程模型
- 建立反馈系统的状态机模型
- 实现机器人大脑，实现状态机控制器

1 介绍

资源:

本实验应与合作伙伴一起完成。每个合作伙伴都应有一台实验室笔记本电脑或一台可以可靠运行的个人笔记本电脑。

- [designLab04工作.py](#): 具有适当导入的模板，用于制作状态机并绘制其输出。
- [smBrainPlotDistSkeleton.py](#): 模板简单的大脑与一个地方为您编写控制器的状态机。
- [第五章](#)课程笔记。

确保将所有代码和图解邮寄给你的搭档。你们俩都需要带着它们参加第一次面试。

设想一个大脑/机器人系统，它可以感知机器人与墙壁之间的距离，并启动机器人电机来移动机器人，使其与墙壁保持一定距离。这是一个简单控制系统的示例，它只有一个输入（与墙壁之间的期望距离）和一个输出（与墙壁之间的实际距离）。我们可以认为控制系统有三个子系统：“控制器”、“工厂”和“传感器”。

认为输入是由用户（例如您）或机器人用来在世界中导航的某些规划系统（例如，将机器人按照某种顺序从一个目标点移动到下一个目标点的状态机）设置的。因此，输入可能会在一段时间内保持不变，然后在一段时间内更改为新值，依此类推。

一些软件和设计实验室包含命令 `athrun 6.01 getFiles`。请忽略此指令：

相同的文件也可以在 [6.01 OCW Scholar](#) 网站上以 .zip 文件的形式获得，标签为 [设计或软件实验室编号] 的代码。

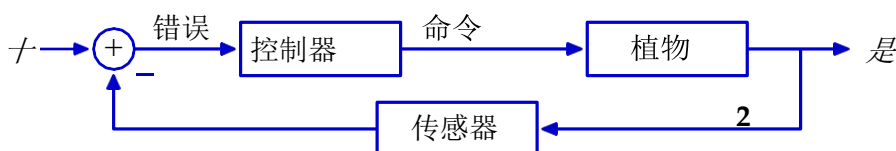


图 1控制系统的结构。对于今天的实验室来说，控制器是机器人的大脑，工厂是被控制的系统的一部分（即机器人的运动系统，其输入是 `io.Action`，其输出是机器人的位置），传感器是机器人的声纳。

通常，我们设计控制器来控制被控对象，使其输出 Y 跟踪某个期望值 X 。例如，在要求机器人与墙壁保持特定距离的情况下，我们希望输出 Y 尽可能接近输入 X 。由于被控对象通常是物理系统，因此对象的输出（例如机器人的位置）不易直接与 X 进行比较。相反，物理输出由传感器测量，传感器的输出（通常是电子的）可以从 X 中减去以确定误差。理想情况下，传感器的输出与被控对象的输出成比例。但更一般地说，传感器会引入其自身的失真、延迟和噪声。

第 4.2.1

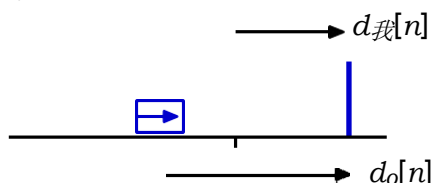
如果你还没有这样做的话，请做一下辅导问题 Wk.4.2.1。

2 墙体探测器的差分方程

客观的: 为墙壁查找器系统开发差分方程模型。

详细指南:

制作一个简单的大脑/机器人系统模型，如下所示。让 $do[n]$ (“o”代表输出) 表示机器人与墙壁之间的当前距离，让 $di[n]$ (“i”代表输入) 表示与墙壁之间的 (当前) 期望距离。还让 v 表示机器人的前进速度。让 $T = 0.1$ 秒表示步骤之间的时间。



当机器人接收到新命令时，我们假设机器人立即改变其速度，然后保持新速度恒定，直到接收到下一个命令（即，机器人加速如此之快，我们可以忽略加速时间）。

检查自己1.给定以下条件，第1步到墙壁的距离是多少？

$$\text{五}[0] = 1$$
$$d_o[0] = 3$$

五[1] =2

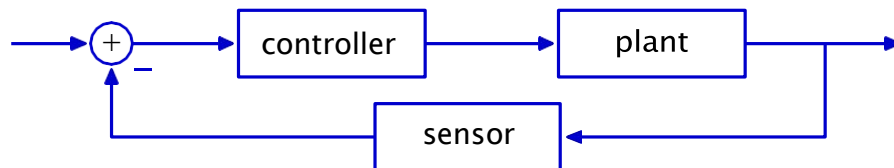
$$d_o[1] =$$

第 4.3.1

将您的答案输入到第 4.3.1 课的第 1 部分并检查。

假设系统具有如下结构图 1 假设传感器测量当前距离 $do[n]$ 并生成感测距离 $ds[n]$ ，该距离等于当前距离延迟一步时间。令 $e[n]$ 表示误差信号，即输入距离 $di[n]$ 和感测距离 $ds[n]$ 之间的差值。在每一步中，控制器命令与误差成比例的前进速度 $v[n]$ ，以便 $v[n] = ke[n]$ 。选择 k ，使得当所需位置位于机器人前方 1 米时，速度为 5 米/秒（请考虑上图显示机器人的位置，以帮助构建 k 的计算）。

检查自己2。完整标记以下系统图。包括 do 、 di 、 ds 、 v 、 e 。



确定差分方程（使用常数 T 和 k ）来关联以下系统组件的输入和输出信号。

- controller

- the model of the plant

- the model of the sensor

第 4.3.1

将这些方程式输入到导师问题 Wk.4.3.1 的第 2、3 和 4 部分中。

结合这些方程，得出一个将 do 与 di 联系起来的差分方程，如下所示：

1. 将差分方程转换为 R 中的算子方程，
2. 用 Di 来求解 Do ，并且
3. 将结果转换回差分方程。

第 4.3.1

将这些方程式输入到导师问题 Wk.4.3.1 的第 5 部分中。

检查1。

第 4.3.2 周：向工作人员解释你的结果。

3 状态机原语和组合器

客观的：

根据系统图表示，构建墙壁查找器系统的状态机模型。

如图 1 的系统图所示，墙壁查找器系统可以建模为原始状态机的组合，使用两个基本结构：`sm.Gain` 和 `sm.R`，因为墙壁寻像器系统是一个“**线性时不变 (LTI) 系统**”。

`sm.Gain` 状态机实际上只是一个纯函数：步骤 `n` 的输出是步骤 `n` 的输入乘以常数 `k`。状态无关紧要。我们将其创建为一种状态机的原因是，我们希望使用 PCAP 的原理将其与其他状态机结合起来，以创建新类型的状态机。

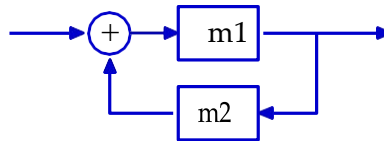
`sm.Gain (SM) :`

```
def init(self, k): self.k = k
def getNextValues(self, state, inp):
    返回 (state, self.k*inp)
```

sm.R 状态机是 Delay 状态机的重命名版本。它在初始化时采用一个值，该值指定机器的初始输出；此后，步骤 n 的输出是步骤 n - 1 的输入。

```
class R(SM)::
    def init (self, v0 = 0):
        self.startState = v0
    def getNextValues (self, state, inp) :
        返回 (inp, state)
```

为了构建 LTI 系统，反馈加法组合将很有用。它需要两台机器并将它们像这样连接起来（请注意，我们在这里使用通用框，如果状态机只是增益，则这些框将是一个三角形，如果状态机是延迟，则将标有 R，或者可能是一些更复杂的反馈回路）：



如果 m1 和 m2 是状态机，那么您可以使用以下方式创建它们的反馈加法组合

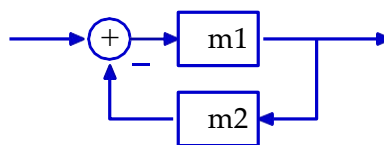
```
newM = sm.FeedbackAdd(m1, m2)
```

现在 newM 本身就是一个状态机。因此，例如，newM = sm.FeedbackAdd(sm.R(0), sm.Gain(1)) 会创建一个状态机，其输出是从步骤 0 到步骤 1 的输入之和不包括当前步骤。您可以通过输入一系列输入来测试它；在下面的示例中，它是数字 0 到 9：

```
>>> newM.transduce (范围 (10))
[0, 0, 1, 3, 6, 10, 15, 21, 28, 36]
```

反馈减法合成是相同的，只是 m2 的输出从输入中减去，得到 m1 的输入。（请注意 m2 输出旁边的减号，因为它被输入到加法器中。）您可以像这样使用它：

```
newM = sm.FeedbackSubtract( (m1, m2)
```



请注意，如果您想在只有一台机器的情况下应用其中一个反馈运算符，则可以使用 sm.Gain(1.0) 或 sm.Wire() 作为另一个参数。

检查自己3.使用增益、延迟和加法器绘制导师问题中第一个系统的系统图第 4.2.1 周。（也就是你来实验室之前做的导师题目）。

检查自己4.使用增益、延迟和加法器绘制导师问题中第二个系统的系统图第 4.2.1 周。

检查自己5.使用增益、延迟和加法器绘制导师问题中第三个系统的系统图第 4.2.1 周。

第 4.3.3 做导师问题 Wk.4.3.3（状态机组成）

检查自己6.使用增益、延迟和加法器绘制控制器的系统图
在墙壁查找系统中。

检查自己7.使用增益、延迟和加法器为墙壁查找器系统中的工厂绘制系统图。

检查自己8.使用增益、延迟和加法器绘制墙壁查找系统中传感器的系统图。

检查自己9.将前三个组件系统连接起来，制作出墙壁探测器系统图。标记所有电线。在控制器、设备和传感器组件周围画出方框。

检查2。 **第 4.3.4 周**向工作人员解释你的系统图。识别墙查找器系统的级联和反馈组合实例。

第 4.3.5 做导师问题 Wk.4.3.5。
将您的代码写入designLab04工作.py。每个函数应返回相应块的状态机实例。使用以下方法测试您的代码：
> 空闲-n
然后在 Tutor 中提交您的代码。

在 $T=0.1$ 且与墙的初始距离为 1.5 米的情况下，尝试不同的增益值。您可以使用 plotD 程序（定义在designLab04工作.py。使用 idle -n。对于给定的增益值 k ，它将绘制到墙壁的距离序列图。

检查自己 10.找到三个不同的值 k ，其中一个距离单调收敛，一个距离振荡并收敛，另一个距离振荡并发散。为每个距离绘制图表 k 值。保存每个图的屏幕截图（请参阅 6.01 网站的“参考”选项卡）。

第 4.3.6 将你发现的收获告诉导师。

4 在模拟机器人上

客观的： 使用状态机实现墙壁查找问题的大脑，如下所述**第 2 部分**。

回想一下，机器人本身就是工厂，因此我们不需要为此编写任何代码。我们已经实现了一个传感器状态机，它输出声纳传感器 3 的值的延迟版本（机器人的传感延迟实际上相对较小）。

您的任务是实现 `Controller` 状态机，该状态机将传感器状态机的输出作为输入，并生成具有 0 旋转速度和适当前进速度的 `io.Action` 实例作为输出。它应该取决于 `dDesired`，即与墙壁的期望距离。通过编辑 `Controller` 类的 `getNextValues` 方法来执行此操作桌面 `/6.01/designLab04/smBrainPlotDistSkeleton.py`。您的控制器应该尝试使声纳传感器 3 的输出等于 0.7，即使传感器 3 没有指向正前方。

检查自己 11. 对于您找到的三个增益 `Check Yourself 10` 中的每一个，在 `wallFinderWorld.py` 世界中运行模拟机器人，并保存情节

检查 3。 **第 4.3.7 周** 比较以下图表**检查自己 10**和**11**。解释它们有何不同，并推测原因。通过电子邮件将您的代码和情节发送给您的合作伙伴。